

무선 센서 네트워크에서 에너지 효율적인 감시·정찰 응용의 클러스터링 알고리즘 연구

공준익*, 이재호*, 강지현**, 엄두섭^o

Energy Efficient Clustering Algorithm for Surveillance and Reconnaissance Applications in Wireless Sensor Networks

Joon-Ik Kong*, Jae-Ho Lee*, Jiheon Kang**, Doo-Seop Eom^o

요 약

다양한 응용에서 사용되고 있는 무선 센서 네트워크(WSN)는 저가의 센서 노드를 구성하기 위해 배터리, 메모리 크기, MCU, RF transceiver 등과 같은 하드웨어에서 제약을 갖고 있다. 특히, 센서 노드의 제한된 에너지는 네트워크 수명과 직접적인 관련이 있기 때문에 네트워크 수명을 연장하기 위한 효율적인 알고리즘이 요구된다. 군 환경에서 침입자를 탐지하기 위한 감시·정찰 응용은 이벤트 구동형(event-driven) 전송 모델로써, 이벤트 발생 빈도가 드물고(rare), 폭발적(bursty), 지역적(local)으로 발생하는 특징이 있다. 이와 같은 응용에서는 Data Aggregation의 장점이 있는 클러스터링 알고리즘을 이용하는 것이 각 노드가 개별적으로 데이터를 전송하는 것 보다 데이터 전송량을 줄여 에너지 효율을 높일 수 있다. 하지만 기존의 클러스터링 알고리즘은 감시·정찰 응용의 이벤트 발생에 대한 특징을 고려하고 있지 않기 때문에 여러 문제가 발생한다. 본 논문에서는 이러한 문제를 개선한 감시·정찰 응용에서의 에너지 효율적인 클러스터링 알고리즘을 제안한다. 이 알고리즘은 타깃을 탐지한 노드들이 각각 Cluster Head Election Window (CHEW)를 생성하여 지역적 경쟁 방식으로 클러스터를 구성하고, 타깃의 이동성을 고려하였다. 시뮬레이션 결과에서는 타깃의 이동에 따라 클러스터가 형성되는 자취를 분석하고, 에너지 효율이 증가되는 것을 증명하였다.

Key Words : WSNs, Clustering Algorithm, Event-Driven, CHEW, Boundary Problem

ABSTRACT

Wireless Sensor Networks(WSNs) are used in diverse applications. In general, sensor nodes that are easily deployed on specific areas have many resource constrains such as battery power, memory sizes, MCUs, RFs and so on. Hence, first of all, the efficient energy consumption is strongly required in WSNs. In terms of event states, event-driven deliverly model (i.e. surveillance and reconnaissance applications) has several characteristics. On the basis of such a model, clustering algorithms can be mostly used to manage sensor nodes' energy efficiently owing to the advantages of data aggregations. Since a specific node collects packets from its child nodes in a network topology and aggregates them into one packet to relay them once, amount of transmitted packets to a sink node can be reduced. However, most clustering algorithms have been designed without considering

※ 본 연구는 민·군겸용기술사업(Dual Use Technology Program)의 지원을 받아 수행되었습니다.

• 주저자 : 고려대학교 전기전자전파공학과 미래정보망 연구실, joonik@korea.ac.kr, 준회원

o 교신저자 : 고려대학교 전기전자전파공학과 미래정보망 연구실, eomds@korea.ac.kr, 종신회원

* 고려대학교 전기전자전파공학과 미래정보망 연구실, izeho75@korea.ac.kr, 정회원

** (주) 센서웨이, kanghead@sensorway.co.kr, 정회원

논문번호 : KICS2012-08-366, 접수일자 : 2012년 8월 20일, 최종논문접수일자 : 2012년 11월 6일

can be reduced. However, most clustering algorithms have been designed without considering characteristics of event-driven delivery model, which results in some problems. In this paper, we propose enhanced clustering algorithms regarding with both targets' movement and energy efficiency in order for applications of surveillance and reconnaissance. These algorithms form some clusters to contend locally between nodes, which have already detected certain targets, by using a method which called CHEW (Cluster Head Election Window). Therefore, our proposed algorithms enable to reduce not only the cost of cluster maintenance, but also energy consumption. In conclusion, we analyze traces of the clusters' movements according to targets' locations, evaluate the traces' results and we compare our algorithms with others through simulations. Finally, we verify our algorithms use power energy efficiently.

I. 서 론

최근 소형의 센서들을 이용한 무선 센서 네트워크(WSNs)의 연구는 산업용, 군사용, 의료용 등의 많은 응용 분야에서 각광받고 있다. 무선 센서 네트워크의 응용에는 건물이나 교량 등의 안전 관리, 군사적인 목표물 추적 및 감시, 의료용 생체공학 등이 있으며, 특히 센서 노드들은 제한된 에너지, 짧은 통신 범위, 낮은 대역폭, 적은 메모리 등의 제한된 자원을 갖기 때문에 응용에 따라 최적의 프로토콜 설계가 요구된다. 즉, 다수의 센서 노드가 주변 환경 정보를 취득하여 주기적으로 보고하는 환경 모니터링 응용과 산불 탐지와 같이 불특정한 시간에 이벤트가 발생하는 응용의 프로토콜 설계는 달라야 한다. 따라서 이러한 응용의 요구에 따라 기존의 프로토콜을 개선하거나 새로운 프로토콜을 설계할 필요가 있다.

무선 센서 네트워크에서 고려해야할 중요한 요소는 에너지 효율이다. 보통 센서 노드들은 사람이 접근하기 힘든 환경에 설치되어 동작하고, 노드의 소형화를 위해 제한된 용량의 배터리를 사용한다. 다시 말해, 센서 노드의 배터리 교환이 어렵기 때문에 센서 네트워크의 에너지 효율은 네트워크의 전체 수명에 직접적인 영향을 미친다. 이러한 문제를 해결하기 위해 효율적인 알고리즘의 연구가 필요하다.

무선 센서 네트워크의 각 노드에서 소비하는 에너지를 세부적으로 살펴보면, 사용자가 원하는 정보를 센싱하는 에너지(Sensing), 그 정보를 노드 내부에서 처리하는 에너지(Data Processing), 그리고 목적지까지 데이터를 전송하는 에너지(Data Transmission)로 구분될 수 있으며, 이 중에서 데이터 전송에 소비되는 에너지가 가장 큰 비중을 차지하고 있다^[1]. 따라서 데이터 전송의 횟수를 줄여 네트워크의 수명을 연장하기 위한 연구가 절실히 필

요하며, 이를 위하여 클러스터링 기술이 최근까지 활발히 연구되고 있다. 일반적으로 계층적 구조를 갖는 클러스터링 알고리즘은 클러스터 헤드와 클러스터 멤버로 구성되며, 클러스터 멤버가 생성한 데이터를 클러스터 헤드가 수집 및 통합(Data Aggregation)하여 중복된 데이터 전송으로 인한 에너지 소모를 줄인다^[2].

무선 센서 네트워크의 데이터 전송 모델은 응용에 따라 연속(continuous), 쿼리 구동형(query-driven), 이벤트 구동형(event-driven) 데이터 전송 모델로 분류할 수 있다^[3]. 먼저, 연속 데이터 전송 모델은 환경 모니터링과 같이 주변 환경의 온도, 습도 등의 정보를 탐지한 데이터를 연속적으로 전송하는 모델이고, 쿼리 구동형 데이터 전송 모델은 Sink 노드의 질의에 따라 각 센서 노드들이 적절한 기능을 수행하고 그 결과에 대한 응답을 전송하는 모델이다. 마지막으로 이벤트 구동형 데이터 전송 모델은 평소에는 동작하지 않고 있다가 이벤트가 발생했을 때 그 정보를 Sink 노드로 데이터를 전송하는 모델이다.

본 논문은 군사 환경을 고려한 감시·정찰 응용으로서, 적의 침입이 예상되는 지역에 성능이 동일한 무선 센서 노드들을 고정된 위치에 설치하여 침입자를 감시 및 추적한다. 이와 같은 감시·정찰 응용은 이벤트 구동형 전송 모델로 다음과 같은 특징을 갖는다. 첫째로 이벤트가 드물게(rare) 발생한다. 침입자를 탐지하기 위해 센서 노드를 설치한 센서 필드에 침입자가 언제 나타날지 모르고, 침입자가 나타나더라도 한순간에 지나가는 특징을 갖는다. 둘째로 센서 노드가 이벤트를 탐지했을 경우 그 주변에서 폭발적(bursty)으로 센싱된 데이터가 발생한다. 즉, 침입자가 나타나지 않는 상황에서 센서 노드들은 침입자의 탐지 메시지를 발생하지 않고 있다가, 침입자가 탐지되면 같은 이벤트에 대해 갑자기 많

은 메시지가 발생된다. 마지막으로 침입자의 탐지는 지역적(local)으로 발생한다. 감시·정찰 응용에서 침입자의 움직임으로 인한 탐지 메시지는 넓은 지역에서 발생하지 않고, 침입자 주변의 일정 범위에서 발생할 확률이 높다.

이와 같이 감시·정찰 응용에서 같은 이벤트를 탐지한 센서 노드들은 데이터가 지역적, 폭발적으로 발생하기 때문에 각 노드가 자신의 데이터를 개별적으로 전송하는 것 보다는 클러스터링 알고리즘의 장점인 Data Aggregation을 이용하면 에너지 효율을 높일 수 있다. 하지만 대표적인 클러스터링 알고리즘인 LEACH^[4]를 비롯한 기존의 고정형(static) 클러스터링 알고리즘은 다음과 같은 문제점이 발생한다. 첫째로, 이벤트가 드물게 발생함에도 불구하고 클러스터를 형성, 유지하기 위한 오버헤드가 발생한다. 둘째로, 침입자의 움직임을 고려하지 않고 센서 노드들을 미리 그룹화하기 때문에 침입자의 움직임에 비효율적이다. 마지막으로 클러스터 경계에서 이벤트가 발생할 경우 불필요한 클러스터의 동작으로 비효율적인 에너지 소모가 발생하는 Boundary Problem^[5]이 발생한다.

본 논문에서는 감시·정찰 응용과 같은 이벤트 구동형 전송 모델에서 타깃의 움직임에 따른 에너지 효율적인 클러스터링 알고리즘을 제안한다. 이벤트가 발생하지 않는 상황에서는 클러스터링 알고리즘이 동작하지 않아 클러스터 유지비용이 발생하지 않고, 움직이는 침입자를 따라 클러스터가 생성 및 소멸되어 불필요한 에너지 소모를 줄이고 Boundary Problem을 해결하였다.

본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 기존의 클러스터링 알고리즘의 문제점을 파악하고, 이에 대한 관련 연구를 요약하였다. 제 3장에서는 감시·정찰 응용에서의 에너지 효율적인 클러스터링 알고리즘을 제안한다. 제 4장에서는 침입자의 이동에 따른 클러스터링 알고리즘의 성능을 비교·분석하였다. 마지막으로 제 5장에서는 본 논문에 대한 결론을 도출하였다.

II. 관련 연구

센서 네트워크의 클러스터링 알고리즘은 조밀하고 무작위로 설치된 센서 노드들을 계층적으로 연결한 클러스터 헤드와 클러스터 멤버로 구성된 클러스터 단위의 통신 방법이다. 이렇게 구성된 클러스터링 알고리즘은 비계층적 네트워크에 비해 다음

과 같은 이유로 에너지 효율이 높다. 첫 번째로 클러스터를 구성하지 않은 네트워크에서는 모든 노드들이 Sink 노드로 각각 데이터를 전송하기 때문에 각 노드에서의 충돌과 재전송으로 인하여 혼잡도가 증가한다. 그러나 클러스터링 알고리즘은 클러스터 헤드가 클러스터 멤버들의 데이터를 수집 및 통합(Data Aggregation)하기 때문에 네트워크의 혼잡도(congestion)을 낮춰 충돌(collision)을 줄일 수 있다. 두 번째로 클러스터 헤드는 자신의 클러스터 멤버들과 통신을 지역적으로 관리할 수 있다. 예를 들어 LEACH와 같이 클러스터 헤드와 클러스터 멤버 간의 통신을 TDMA를 이용한다면 데이터 전송 시 충돌을 줄여 재전송 횟수를 줄일 수 있고, 할당된 Time Slot에만 데이터 전송이 이루어지므로 에너지 소모를 줄일 수 있다.

대표적인 클러스터링 알고리즘은 LEACH, HEED^[6] 등이 있다. LEACH는 set-up phase와 steady-state phase로 구성된 round가 반복되어 수행된다. set-up phase는 클러스터가 구성되는 단계로서, 임의의 노드가 클러스터 헤드로 선정되고, 각 클러스터 헤드를 중심으로 클러스터 멤버들을 구성한다. 그리고 에너지 소모의 균형을 맞추기 위해 매 round의 set-up phase 마다 새로운 클러스터 헤드를 선정한다. steady-state phase는 실질적 데이터 전송 단계이다. HEED는 각 노드가 에너지 잔량과 이웃 노드와의 가까움 정도를 이용한 확률로 클러스터 헤드를 선출한다. 그 후 클러스터 헤드의 부하를 줄이기 위해 클러스터를 재구성하는 과정을 반복한다.

이와 같은 기존의 클러스터링 알고리즘은 에너지 효율을 위해 주기적으로 클러스터를 재구성하여 센서 노드의 에너지를 균등하게 소비하도록 하였지만, 본 논문에서 고려하는 이벤트 구동형(event-driven) 전송 모델의 감시·정찰 응용에서 움직이는 타깃을 고려하지 않고 클러스터가 구성되기 때문에 오히려 비효율적인 에너지 소모를 초래한다. 따라서 이러한 문제를 개선하기 위해 이벤트 구동형 전송 모델에 맞도록 기존의 클러스터링 알고리즘을 개선하는 연구가 진행되고 있다.

먼저 HCTT^[5] 알고리즘은 하이브리드(hybrid) 클러스터링으로 고정형(static) 클러스터링과 동적(dynamic) 클러스터링을 혼합한 방법이다. 크게는 고정형 클러스터링을 유지하면서 타깃에 대한 정보를 수집하고, 타깃이 움직여 고정형 클러스터의 경계에 위치할 경우 Boundary Problem을 해결하기 위해 동적 클러스터를 형성한다. Boundary Problem

은 클러스터 경계에서 이벤트 발생 시 주위의 여러 클러스터가 개별적으로 정보 수집하여 같은 정보를 공유하지 않는 문제와 이벤트 발생 시 주위의 모든 클러스터가 동작하여 비효율적인 에너지가 사용되는 문제이다. 만약 타깃이 고정형 클러스터의 경계를 따라 움직일 경우 타깃의 움직임에 따라 클러스터가 형성되는 동적 클러스터링과 같아진다.

Event-Driven Clustering Routing Algorithm^[7]에서는 미리 정해진 Gateway 노드들이 주변 노드들의 에너지 잔량 정보를 수집하고, 에너지 잔량이 높은 노드가 클러스터 헤드로 선정될 수 있도록 하는 중요한 역할을 수행한다. 데이터 전송에 있어서는 클러스터 헤드가 수집 및 통합한 데이터를 Gateway 노드를 통해 Base Station으로 전송한다. 하지만, Gateway 노드가 장애가 발생했을 경우에는 네트워크 전체에 문제가 발생할 수 있고, Gateway 노드가 주변 노드들의 에너지 잔량 정보를 수집하기 위한 control overhead가 발생한다.

AEEC^[8] 알고리즘은 LEACH 알고리즘을 개선하여 이벤트 구동형 전송 모델에 맞게 수정하였다. 에너지 잔량이 높은 노드들이 클러스터 헤드와 주위의 에너지 정보를 수집하는 elector로 선정된다. elector는 네트워크 수명을 연장하기 위해 다음 라운드의 클러스터 헤드와 elector를 선출하는 과정을 반복적으로 수행하여 에너지 사용의 균형을 맞춘다. 이 알고리즘은 이벤트 구동형(event-driven) 데이터 전송 모델에서 고정형 클러스터링 알고리즘에 비해 에너지 효율을 높였지만, 주변의 에너지 잔량을 획득하기 위한 control overhead가 발생한다.

본 논문에서는 이벤트 구동형 전송 모델인 감시·정찰 응용에서 타깃의 움직임에 따라 클러스터를 생성 및 소멸하여 기존의 알고리즘 보다 에너지 효율적인 클러스터링 알고리즘을 제안한다. 제안하는 감시·정찰 응용의 클러스터링 알고리즘은 다음과 같은 특징이 있다. 기존의 클러스터링 알고리즘의 장점인 Data Aggregation을 그대로 유지하여 데이터 충돌 및 에너지 소모를 효율적으로 관리할 수 있고, 침입자가 탐지되지 않는 상황에서는 클러스터를 형성하지 않기 때문에 클러스터 유지를 위한 control overhead가 발생하지 않아 유지비용 측면에서도 장점을 갖는다. 또한 침입자의 움직임에 따라 클러스터를 생성 및 소멸하기 때문에 침입자의 움직임에 민감하게 반응할 수 있고, 타깃을 중심으로 클러스터가 형성되기 때문에 클러스터 경계에서 이벤트 발생 시 나타나는 Boundary Problem을 해결할 수

있다.

다음으로, 클러스터 헤드를 선출하는 과정에서 기존의 이벤트 구동형 클러스터링 알고리즘과 달리 지역적 경쟁방식으로 클러스터 헤드를 선출하기 때문에 Gateway^[7] 노드나 Elector^[8]와 같이 임의의 노드가 주변 노드의 에너지 잔량을 확인하기 위한 control overhead가 발생하지 않는다. 그리고 제안하는 알고리즘에서는 클러스터링 알고리즘에서 중요하게 고려해야하는 클러스터 헤드의 에너지 소모 집중 문제를 해결하기 위해, 각각의 센서 노드들은 자신의 에너지 잔량 정보를 활용하여 클러스터 헤드 선출 과정을 진행한다. 따라서 주변 노드들과 비교하여 에너지 잔량이 풍부할수록 클러스터 헤드로 선출될 확률이 높다. 또한, 본 논문에서 고려하고 있는 감시·정찰 응용은 타깃의 움직임이 있기 때문에 클러스터의 잦은 생성 및 소멸 과정이 반복되어 클러스터 유지 시간이 짧고, 클러스터 헤드의 임무 수행 시간이 짧아 에너지 소모의 집중화 문제가 크지 않다. 이로써 제안하는 클러스터링 알고리즘은 센서 노드들의 균형 있는 에너지 사용으로 네트워크 수명을 연장할 수 있다.

III. 감시·정찰 응용의 클러스터링 알고리즘

3.1. 문제점 정의

성능이 동일한 센서 노드로 구성된 무선 센서 네트워크는 많은 응용 분야에서 활용된다. 그리고 각 응용에 따라 무선 센서 네트워크의 성능을 극대화하기 위한 알고리즘 연구가 진행되고 있다. 그중에서도 클러스터링 알고리즘은 각 센서 노드들을 클러스터 헤드와 클러스터 멤버로 구성함으로써, 온도·습도와 같은 환경 모니터링 응용과 같이 고정된 각 센서 노드들이 같은 데이터를 중복으로 전송하여 발생하는 데이터 충돌(collision) 및 에너지 소모를 효율적으로 처리할 수 있다. 하지만 클러스터 헤드로 선정되는 노드는 클러스터 멤버들의 데이터를 수집 및 통합(Data Aggregation)하기 때문에 다른 노드들에 비해 에너지 소모가 집중되는 문제가 발생하게 된다. 이와 같은 문제를 해결하기 위해 LEACH와 같은 기존의 클러스터링 알고리즘은 주기적으로 에너지 잔량이 많은 노드가 클러스터 헤드로 선정할 수 있도록 하였다.

그러나 위와 같은 클러스터링 알고리즘은 감시·정찰 응용과 같은 이벤트 구동형(event-driven) 전송 모델에서는 다음과 같은 문제가 발생한다. 첫째로,

이벤트가 발생하지 않아도 클러스터 유지비용이 발생한다. 감시·정찰 응용에서는 침입자가 언제 나타날지 모른다. 이렇게 언제 발생할지 모르는 이벤트를 탐지하기 위해 클러스터를 항상 유지해야하고, 이벤트가 없는 환경에서도 불필요한 control message를 서로 교환하기 때문에 에너지 효율이 매우 낮다. 둘째로, 미리 클러스터를 구성하기 때문에 침입자의 움직임에 비효율적이다. 미리 구성된 클러스터들은 예측하기 힘든 침입자의 움직임을 고려하지 않아 불필요한 노드들이 동작할 수 있다. 셋째로, 침입자가 클러스터 경계에서 탐지될 경우 Boundary Problem이 발생한다. Boundary Problem은 클러스터 경계에서 이벤트 발생 시 주위의 여러 클러스터가 개별적으로 정보 수집하여 같은 정보를 공유하지 않아 주위의 모든 클러스터가 동작하여 비효율적인 에너지가 사용되는 문제이다. 따라서 감시·정찰 응용과 같이 데이터가 폭발적(burst), 지역적(local)으로 발생하는 응용에서 Boundary Problem은 심각하다.

이와 같은 문제를 해결하기 위해 감시·정찰 응용과 같이 이벤트 구동형 전송 모델에 적합하도록 개선된 클러스터링 알고리즘의 연구가 필요하다. 본 논문에서는 타깃의 움직임에 따라 동작하는 에너지 효율적인 클러스터링 알고리즘을 제안하여 다음과 같이 기존 문제를 개선하였다. 첫째로, 타깃의 움직임이 없는 경우 클러스터를 구성하지 않고 오직 타깃이 탐지될 경우에만 클러스터가 동작하기 때문에 각 센서 노드 간 불필요한 메시지를 전송하지 않아 control overhead가 발생하지 않는다. 둘째로, 기존과 같이 미리 클러스터를 구성하지 않고 그림 1과 같이 타깃의 움직임에 반응하여 클러스터를 생성 및 소멸 과정을 반복하기 때문에 불필요한 노드의 동작을 최소화하고 에너지 효율을 높일 수 있다. 마지막으로, 클러스터 구성 시 타깃을 탐지한 각 센서 노드들은 지역적 경쟁으로 클러스터 헤드를 선정하기 때문에 타깃과 가까운 센서 노드가 클러스터 헤드로 선정될 확률이 높고, 그것을 중심으로 클러스터가 형성되어 클러스터 경계에서 이벤트 발생 시 문제가 되는 Boundary Problem을 최소화할 수 있다.

제안하는 클러스터링 알고리즘은 감시·정찰 응용에서 타깃이 탐지하여 발생하는 데이터가 드물게(rare) 발생하지만, 타깃이 나타나면 지역적(local)으로 폭발적(bursty)인 데이터가 발생하는 것을 가정하고 있다. 따라서 타깃을 탐지하지 않는 일반적인

상황에서는 사용자의 요구사항(Query)에 대한 응답을 위한 라우팅 경로가 미리 구성되어 있다고 가정하고, 타깃이 발생하면 에너지 효율을 높이기 위해 클러스터링 알고리즘이 동작한다. 즉, 클러스터링 알고리즘은 필요시에만 동작하기 때문에 Add-on 형태로 클러스터링 알고리즘이 동작하고 클러스터 헤드가 수집한 데이터는 미리 구성되었던 라우팅 경로를 통해 목적 노드까지 전달된다. 본 논문에서는 트리 토폴로지를 미리 구성하고, Add-on 형태로 클러스터링 알고리즘을 적용하여 실험하였다.

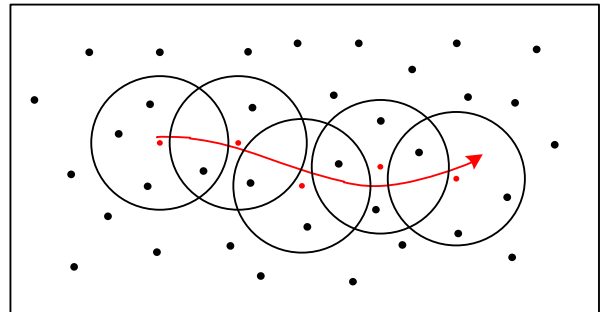


그림 1. 감시·정찰 응용의 클러스터링 알고리즘
Fig. 1. The clustering algorithm for surveillance and reconnaissance applications

3.2. 클러스터 구성 과정

먼저, 무선 센서 네트워크에 사용되는 센서 노드들의 통신 거리는 약 100m이고, 타깃을 탐지하기 위해 센서 노드에 설치된 IR 센서 등의 센싱 거리는 약 5~20m이다. 이와 같이 보통의 센서 노드는 통신 거리가 센싱 거리 보다 크기 때문에 제안하는 클러스터링 알고리즘에서도 통신 거리가 센싱 거리 보다 크도록 설정한 후 시뮬레이션을 이용해 실험하였다.

본 논문에서 제안하는 감시·정찰 응용의 클러스터링 알고리즘은 동일 성능의 센서 노드가 상황에 따라 각각 클러스터 헤드와 클러스터 멤버로 동작한다. 각 클러스터는 기존의 알고리즘과 같이 고정된 형태로 구성되는 것이 아니라, 타깃을 탐지한 센서 노드들이 지역적 경쟁으로 타깃의 움직임에 따른 최적의 클러스터를 동적으로 구성한다. 이때 경쟁에서 승리한 노드는 클러스터 헤드가 되고, 그것을 중심으로 주변의 센서 노드들은 클러스터 멤버가 된다.

제안하는 클러스터링 알고리즘은 타깃을 탐지한 센서 노드들이 지역적 경쟁 방식으로 클러스터 구

성하여 에너지 효율을 높이기 위해, MAC layer에서 주변 노드들과 패킷 충돌을 회피하기 위한 방법 중 IEEE802.11 알고리즘의 CW(Contention Window)와 IFS(Inter Frame Space) 동작 과정^[9]에서 아이디어를 얻어 Routing Layer에 적용한 새로운 클러스터링 알고리즘을 연구 하였다. 먼저, 임의의 센서 노드가 타깃을 탐지하면, 그림 2와 같은 과정으로 클러스터가 구성된다. 각 센서 노드는 서로 경쟁하기 위해 타깃과의 거리(Dtarget)와 배터리 잔량(Ebattery)를 이용한 식(1)로 서로 다른 크기의 Cluster Head Election Window(CHEW)를 독립적으로 생성한다. 식(1)의 첫 번째 항은 정수 부분으로 CHEW의 최대 크기를 결정하는 역할을 하고, 두 번째 항은 0과 1 사이의 소수 부분으로 자신의 CHEW가 주변 노드들의 CHEW와 동일할 경우 동시 전송으로 인한 패킷 손실을 막는 역할을 한다. CHEW는 식(2)와 같이 Cluster Head Election Time(CHET)를 계산하기 위한 중요한 요소로서, 타깃을 탐지한 노드들은 CHET를 이용해 클러스터 헤드라 되기 위한 경쟁을 시작한다. 즉, 타깃과 가깝고 배터리 잔량이 많은 노드일수록 CHEW의 크기가 작게 설정되고, 클러스터 헤드로 선출될 확률이 높다. 이러한 방법은 기존의 알고리즘에서 특정 노드가 주변 노드들의 에너지 잔량을 파악하기 위한 control overhead를 제거하고, CHEW를 이용한 지역적 경쟁 방식으로 클러스터 헤드를 선정하기 때문에 에너지 효율이 높다.

$$CHEW = INT\left(\frac{D_{target}}{E_{battery}}\right) + Random(), \quad (1)$$

where $0 \leq Random() \leq 1$

$$CHET = CHEW * slottime \quad (2)$$

클러스터 헤드 선정을 위해 타깃을 탐지한 노드들이 생성한 CHET는 일종의 지연시간으로써, 식(2)와 같이 각 노드가 생성한 CHEW와 노드를 설치할 때 사용자에게 의해 고정값으로 설정되는 slottime의 곱으로 나타낸다. 다시 말해, 각 노드가 설정한 CHET의 시간이 지났다면 CH_ADV (Cluster Head Advertisements) 메시지를 broadcast로 주변 노드들에게 전송하여 자신이 클러스터 헤드의 후보임을 알린다. 따라서 빠른 시간에 CH_ADV 메시지를 전송한 노드가 경쟁에서 승리하고 클러스터 헤드가 된다.

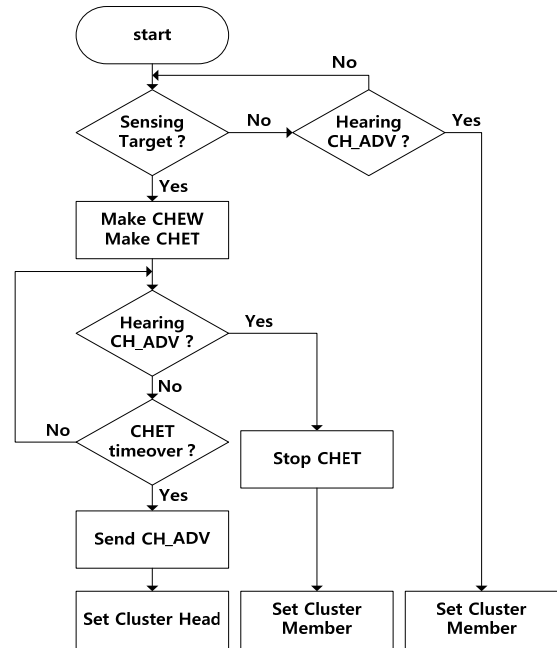


그림 2. 클러스터 헤드 및 멤버 선정의 순서도
Fig. 2. The procedure for determining either a cluster head or its member

예를 들어, 임의의 시간에 그림 3과 같이 타깃을 탐지한 노드가 4개 있다고 가정하자. 이 노드들은 식(1)을 통해 서로 다른 크기의 CHEW를 생성한다. 즉, A의 CHEW는 2, B의 CHEW는 3, C의 CHEW는 4, D의 CHEW는 5로 생성되었다. CHEW의 크기가 작은 것은 CHET의 시간이 짧은 것으로, 그림 3에서 노드 A의 CHET가 가장 짧기 때문에 경쟁 노드들에 비해 빠른 시간에 CH_ADV 메시지를 전송할 수 있고 다른 노드들과 경쟁에서 승리할 수 있다. 따라서 그림 3에서는 CHEW의 크기가 2인 노드 A가 클러스터 헤드로 선정된다.

클러스터 멤버는 클러스터 헤드가 송신한 CH_ADV 메시지를 통해 결정된다. 클러스터 헤드는 지역적 경쟁에서 승리한 하나의 노드이므로 경쟁에 참여했던 노드뿐만 아니라 클러스터 헤드의 통신 범위 안에 있는 모든 노드들도 클러스터 멤버가 되어 하나의 클러스터가 구성된다. 다시 말해, 서로 경쟁했던 센서 노드들은 자신의 CHET 이전에 다른 노드가 송신한 CH_ADV 메시지를 수신했을 때, 그림 3과 같이 수행 중인 자신의 CHET를 멈춰 클러스터 헤드 선출 과정을 포기하고 클러스터 멤버가 된다. 또한, 센서 노드의 통신 범위가 센싱 범위보다 넓기 때문에 이벤트를 탐지를 하지 않아 경쟁에 참여하지 않은 노드들도 CH_ADV 메시지를

수신하고 클러스터 멤버가 되어 클러스터를 구성한다. 왜냐하면 이 노드들은 타깃이 움직일 경우 짧은 시간에 타깃을 탐지할 확률이 높기 때문이다.

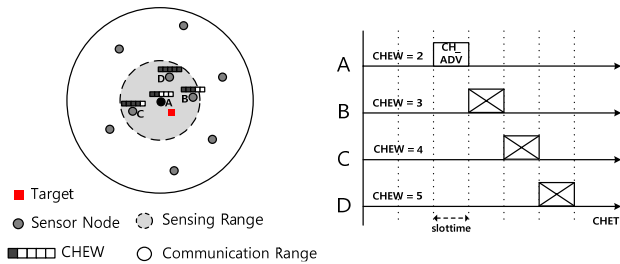


그림 3. 클러스터 헤드 선정 과정
Fig. 3. The process of a cluster-head's selection

이와 같이 CHEW를 이용한 클러스터를 구성 방법은 기존의 방법처럼 특정 노드가 클러스터 헤드 선정과정에 개입하는 것이 아니라 센서 노드들 간에 지역적으로 경쟁하여 승리한 노드가 한번의 CH_ADV 메시지 전송으로 클러스터를 구성하기 때문에 control overhead를 줄여 에너지 효율을 높이는 장점이 있다. 하지만, CH_ADV 메시지를 한번 보내기 때문에 패킷 충돌, 손실 등으로 인한 통신 장애로 주변 노드들이 CH_ADV 메시지를 수신하지 못하는 경우가 발생할 수 있다. 이와 같은 문제는 다음과 같이 나누어 볼 수 있다.

첫째로는 클러스터 헤드가 CH_ADV 메시지를 송신하였지만 경쟁에 참여했던 노드 중 우선순위가 낮은 노드가 이 메시지를 수신하지 못하는 경우이다. 이와 같은 경우 우선순위가 낮은 노드는 기존의 방법과 같이 자신의 CHEW를 생성하여 클러스터 헤드 선정 과정을 그대로 수행하고, 자신의 CH_ADV 메시지를 broadcast 한다. 하지만 이와 같은 상황에서는 이미 클러스터가 구성되어 있기 때문에 오직 클러스터 헤드만이 우선순위가 낮은 노드에게 CH_ADV 메시지에 대한 응답으로 클러스터가 이미 형성되었음을 확인시키고 클러스터 멤버로 전환할 수 있도록 한다. 또한, 클러스터 헤드의 응답 메시지를 통신장애로 인해 수신하지 못한 노드는 새로운 클러스터를 형성할 수 있다. 이 경우에는 단독 클러스터를 형성한 후 클러스터 헤드의 메시지를 overhearing하여 클러스터의 존재 인지하고 클러스터 멤버로 전환할 수 있다. 왜냐하면 클러스터의 크기는 센서 노드의 통신 범위로 결정되기 때문에 하나의 클러스터에는 하나의 클러스터 헤드만 존재한다.

둘째로는 타깃을 탐지 하지 않아 클러스터 헤드 선출 경쟁에 참여하지 않은 노드들이 클러스터 헤드의 CH_ADV 메시지를 수신하지 못하는 경우이다. 이 노드들은 타깃의 이동성으로 인하여 곧 타깃을 탐지할 확률이 높을 뿐, 당장 클러스터 구성에 참여하지 않아도 큰 문제가 발생하지 않기 때문에 별도의 동작을 하지 않는다. 하지만 타깃이 이동하여 이 노드의 센싱 범위에 들어올 경우에는 앞선 과정과 동일하게 자신의 CHEW를 생성하여 CH_ADV를 송신한 후, 기존의 클러스터에 귀속되어 클러스터 멤버가 되거나 새로운 클러스터를 형성한다.

3.3. 클러스터의 이동

감시·정찰 응용에서는 고정형 클러스터링의 단점을 극복하기 위해 타깃의 움직임에 따른 효율적인 클러스터링 알고리즘이 요구된다. 제안하는 클러스터링 알고리즘은 지역적 경쟁 방식으로 클러스터 헤드를 선출하고, 이를 중심으로 형성된 클러스터는 타깃의 움직임을 고려하여 생성 및 소멸을 반복한다. 즉, 타깃이 이동할 경우 타깃의 정보를 지속적이고 효율적으로 획득하기 위해 새로운 클러스터를 생성하고, 불필요하게 된 클러스터는 소멸하여 에너지 효율을 높인다.

그림 4는 타깃의 이동으로 새로운 클러스터를 구성하는 과정을 보여준다. 이때 통신 범위는 R , 센싱 범위는 r 로 정의한다. 먼저, 그림 4(a)는 현재 구성되어 있는 클러스터에서 어느 영역으로 타깃이 이동하면 새로운 클러스터가 형성되는지를 나타낸다. 새로운 클러스터의 형성 시점은 타깃이 어느 방향으로 움직일지 예측할 수 없기 때문에 클러스터 외부의 노드가 타깃을 처음으로 탐지하는 순간으로 정의된다. 즉, 그림 4(a)와 같이 현재 클러스터의 외부 경계로부터 센싱 범위 r 만큼 떨어진 도넛 모양의 빗금 영역은 타깃이 아직 클러스터 내부에 존재하지만 현재 클러스터를 구성하고 있는 노드가 아닌 외부의 노드가 타깃을 탐지할 수 있는 최대 영역이며, 타깃이 이 영역으로 이동할 경우 새로운 클러스터를 구성하기 위한 과정이 시작된다. 그림 4(b)는 타깃이 빗금 영역에 들어와 클러스터 외부 노드들이 타깃을 탐지했을 때, 이 노드들이 서로 경쟁하여 새로운 클러스터 헤드를 선출하는 과정을 나타낸다. 다시 말해, 타깃의 이동으로 클러스터 외부에 있는 노드 A와 노드 B가 동시에 타깃을 탐지했다고 가정하자. 노드 A와 노드 B는 새로운 클러

스터 헤드의 후보로서 각각 자신의 CHEW를 생성하여 새로운 클러스터 헤드가 되기 위해 서로 경쟁한다. 이 경쟁에서 승리한 노드는 클러스터 헤드가 되고, CH_ADV 메시지를 통해 새로운 클러스터 멤버들을 구성한다. 이러한 과정이 반복되면서 클러스터는 타깃의 이동에 따라 지속적으로 생성 된다.

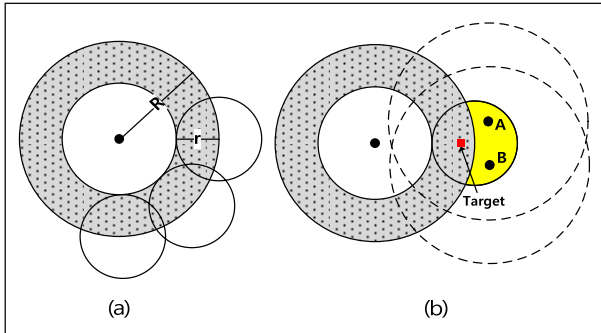


그림 4. 새로운 클러스터 형성
Fig. 4. New cluster formation

3.4. 클러스터의 중첩 및 소멸

타깃의 움직임에 따라 클러스터가 형성될 경우 그림 5와 같이 서로 다른 클러스터 영역에 중첩되는 클러스터 멤버가 발생한다. 이것은 새롭게 선출된 클러스터 헤드의 탐지 영역이 기존 클러스터 내에 존재하기 때문이며, 이렇게 중첩된 클러스터 멤버 노드들은 다음과 같이 동작한다. 먼저, 중첩된 노드는 새롭게 형성되는 클러스터의 멤버로 링크를 유지하되, 기존의 클러스터 멤버의 역할에 우선순위를 둔다. 만약 기존 클러스터에 대한 링크 손실 등의 장애가 발생했을 때 새로운 클러스터의 링크를 유지하기 때문에 정보 손실을 방지할 수 있다. 다음으로 중첩된 노드들은 타깃의 정보를 서로 다른 클러스터에 공유할 수 있다. 타깃을 추적 및 감시하는 응용에서는 타깃의 과거 움직임에 대한 정보가 필요하다. 이때 중첩된 노드들은 타깃의 과거 정보를 새로운 클러스터와 공유할 수 있다.

타깃이 기존 클러스터 영역을 완전히 벗어날 경우, 기존에 형성되었던 클러스터는 더 이상 타깃을 탐지하지 못하게 되며, 이 경우 기존의 클러스터는 유지비용 측면이나 에너지 효율 측면에서 소멸되어야 한다. 따라서 클러스터를 구성하고 있는 모든 노드가 일정한 시간 동안 타깃을 탐지하지 않을 경우 클러스터 헤드는 클러스터 해지 메시지를 클러스터 멤버들에게 전송함으로써 클러스터를 소멸시킨다.

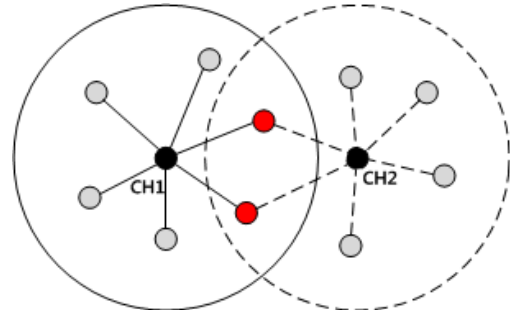


그림 5. 인접한 클러스터 간의 중복된 멤버
Fig. 5. In case of duplicated members between clusters

IV. 성능 평가

4.1. 시뮬레이션 환경

본 장에서는 제안된 감시·정찰 응용의 클러스터링 알고리즘 성능을 평가하기 위해 NS-2^[10]를 활용하여 표 1과 같이 네트워크 환경을 설정하고 시뮬레이션 하였다. 센서 노드들은 20m 간격의 격자 구조로 배치하고, 타깃이 이 구역을 다양한 속도로 움직이는 것을 가정하여 클러스터의 생성 및 소멸 과정을 확인하고 성능을 분석하였다. 타깃의 이동 속도는 5m/s ~ 20m/s로 사람의 이동 속도에서부터 동물이 빠르게 달리는 속도까지 시뮬레이션 하였다.

실험 내용은 다음과 같다. 첫 번째로 타깃의 이동에 따라 형성되는 클러스터의 자취를 분석하였고, 두 번째로 네트워크 크기에 따라 형성되는 클러스터를 분석하고, 네트워크 크기가 10x10에서 타깃의 다양한 이동 속도에 따른 성능을 분석하였다.

표 1. 시뮬레이션 환경
Table 1. Simulation environments

Parameter	Value
Simulation Tool	NS-2.33
Network Topology	10x10, 15x15, 20x20, 25x25
Distance between nodes	20m
Target Velocities	5/10/15/20 m/s
Traffic Type	CBR
Data Length	100 bytes
Transmission Range	30 m
Sensing Range	15 m
slottime	20µs

4.2. 시뮬레이션 결과 및 성능 분석

4.2.1. 타겟의 이동에 따른 클러스터 형성 자취 분석

제안하는 클러스터링 알고리즘은 타겟의 움직임에 따라 클러스터가 생성 및 소멸 과정을 반복한다. 타겟의 움직임이 탐지되지 않았을 경우에는 센서 노드들이 클러스터를 형성하지 않아 불필요한 에너지 소모를 줄인다. 그리고 센서 노드가 타겟을 탐지했을 경우에는 제안하는 알고리즘에 따라 각각 CHEW를 생성한다. 그 후 각 노드 간 경쟁을 통해 클러스터 헤드를 선출하고 클러스터를 구성한다. 또한 동일한 과정으로 타겟의 이동에 따라 클러스터가 형성되고, 과거의 클러스터는 소멸시켜 에너지 소모를 줄인다. 그림 6은 타겟이 대각선 방향으로 움직일 때 형성되는 클러스터의 자취를 나타낸 그림이다. 타겟의 이동 경로에 근접한 노드들은 타겟의 정보를 다량으로 전송할 확률이 높다. 하지만 제안하는 알고리즘에서는 이러한 노드들이 클러스터로 구성되어 중복 데이터 전송의 감소로 에너지 소모를 줄일 수 있다.

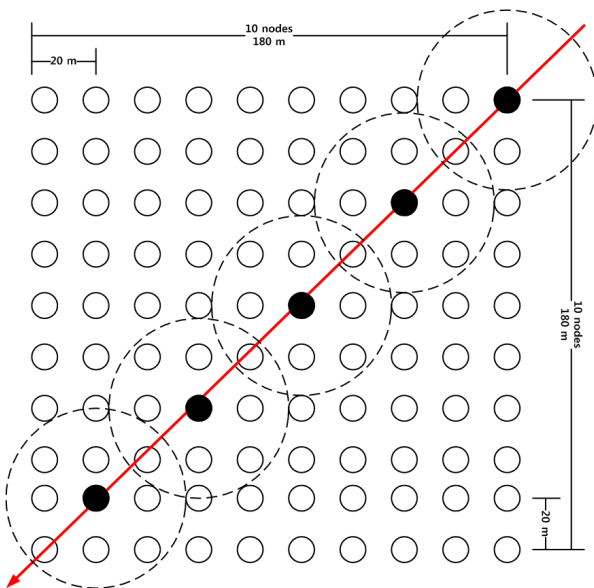


그림 6. 타겟의 이동에 따른 클러스터 자취
Fig. 6. The trace of clusters along a target's movement

타겟의 이동에 따라 클러스터가 형성되려면 클러스터 구성 시간은 타겟의 움직이는 속도보다 빨라야 한다. 제안하는 클러스터링 알고리즘은 타겟을 탐지했을 때, 주변 노드들과 정보를 공유하지 않고 개별적으로 생성한 CHEW를 이용해 경쟁하기 때문에 클러스터 헤드 선정 과정에서 메시지 교환으로

인한 시간 지연이 발생하지 않는다. 또한 클러스터 멤버들도 클러스터 헤드가 전송한 한번의 CH_ADV 메시지를 통해 구성되기 때문에 빠르게 클러스터가 형성된다.

그리고 센서 노드가 설치된 환경에서 타겟의 움직임에 따라 클러스터가 생성과 소멸을 반복하기 때문에, 타겟을 탐지하기 위한 최소의 클러스터가 동작하여 불필요한 에너지 소모를 줄일 수 있다. 즉, 그림 7과 같이 타겟이 이동할 때 클러스터의 동작 상태를 살펴보면, 먼저 타겟이 A 위치에 있을 경우 오직 클러스터 1이 타겟의 정보를 수집하고, 타겟이 B 위치로 이동하면 타겟의 이동에 따른 정보 전송의 연속성을 위해 새로운 클러스터인 클러스터 2를 형성한다. 다음으로 타겟이 C 위치로 이동할 경우 타겟의 정보는 클러스터 2에서 수집하고, 클러스터 1은 타겟을 탐지할 수 있는 범위를 벗어났기 때문에 소멸시켜 불필요한 에너지 소모를 막는다. 마찬가지로, 타겟이 D 위치와 E 위치로 이동할 경우, 앞의 과정과 동일한 과정으로 각각 클러스터 3을 생성하고 클러스터 2를 소멸한다.

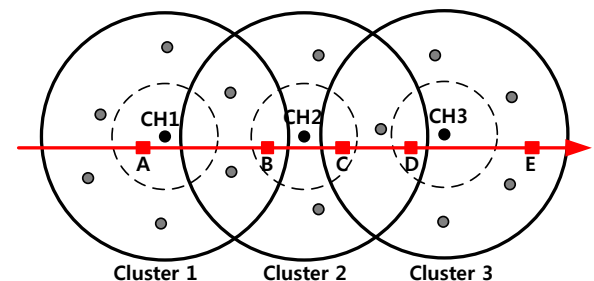


그림 7. 타겟의 이동에 따른 클러스터 생성 및 소멸 과정
Fig. 7. The process of cluster's construction and extinction during a target's movement

4.2.2. 네트워크의 크기 및 타겟의 이동 속도에 따른 성능 분석

그림 8은 타겟이 대각선 방향으로 움직이고 이동 속도를 10m/s로 고정하여 네트워크의 크기 변화에 따라 형성되는 클러스터를 분석하였다. 네트워크의 크기가 커질수록 생성되는 클러스터의 수도 선형적으로 증가하는 것을 확인할 수 있다. 즉, 이동하는 타겟이 센서 노드가 설치된 지역을 오래 머물수록 형성되는 클러스터 수가 증가한다. 타겟이 한 곳에 머물 경우에는 하나의 클러스터 동작으로 타겟의 정보를 획득할 수 있지만, 타겟이 계속적으로 움직일 경우에는 타겟의 움직임에 따라 클러스터가 형

성되기 때문이다. 하지만, 앞서 언급한 내용처럼 한 시점에 동작하는 클러스터는 최대 두 개이고, 기존의 알고리즘처럼 주기적으로 발생하는 control overhead가 없기 때문에 제안하는 클러스터링 알고리즘이 에너지 측면에서 보다 효율적이다.

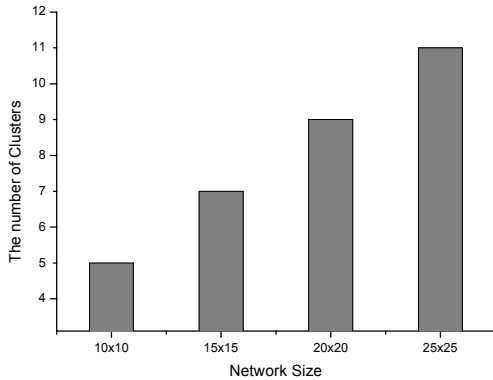


그림 8. 네트워크 크기에 따라 형성되는 클러스터 수
Fig. 8. The number of clusters formed as per network size

그림 9에서는 타깃의 이동 속도에 따른 평균 소비 에너지를 시뮬레이션 하였다. 먼저, 제안하는 클러스터링 알고리즘이 에너지 소모가 가장 적은 것을 확인할 수 있다. 그리고 LEACH가 대체적으로 에너지 소모가 많은 것을 확인할 수 있다. 제안하는 클러스터링 알고리즘은 타깃의 움직임에 따라 클러스터를 생성 및 소멸하기 때문에 최소의 클러스터가 동작하고, Boundary Problem이 개선되어 불필요한 클러스터가 동작하지 않기 때문에 에너지 효율을 높였다는 것을 확인할 수 있다. 하지만, LEACH는 Boundary Problem과 클러스터를 유지하기 위한 control overhead가 있기 때문에 제안하는 알고리즘 보다 많은 에너지 소모를 보인다. 그리고 타깃의 이동 속도가 느릴 경우에는 세 알고리즘의 평균 에너지 소모량의 차이가 크게 나타나지만, 타깃의 이동 속도가 빠를 경우에는 큰 차이가 없는 것을 확인할 수 있다. 이것은 타깃의 이동 속도가 빠르면 클러스터가 동작하는 과정은 동일하지만, 센서 노드가 타깃을 탐지하여 Sink 노드로 보고하는 데이터양이 줄어들어 Data Aggregation의 효과가 줄어들기 때문이다.

그림 10에서는 타깃의 이동 속도에 따른 송·수신 패킷의 비율을 시뮬레이션 하였다. 실험 결과는 타깃의 이동 속도에 관계없이 대체적으로 유사한 패

킷 송수신 비율을 나타내는데, 제안하는 알고리즘은 약 1.6배, LEACH는 약 3배, non-clustering에서는 약 5배의 비율이 나타난다. 이것은 제안하는 클러스터링 알고리즘이 타깃을 중심으로 최소의 클러스터를 형성하기 때문에 LEACH에서 발생하는 Boundary Problem의 불필요한 클러스터의 동작을 줄여 Sink 노드로 보고하는 메시지의 양을 줄였기 때문이다. 따라서 제안하는 클러스터링 알고리즘으로 불필요한 데이터 전송을 막을 수 있고, 이로써 에너지 효율을 높일 수 있다.

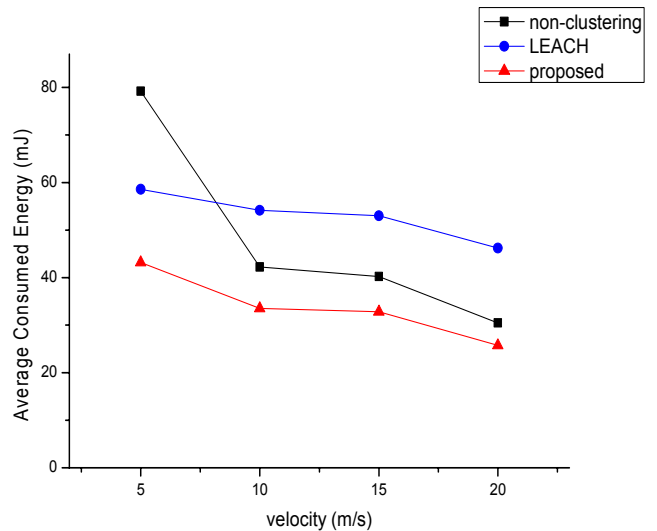


그림 9. 타깃의 이동 속도에 따른 평균 소비 에너지
Fig. 9. The average energy consumptions vs. velocities for a target

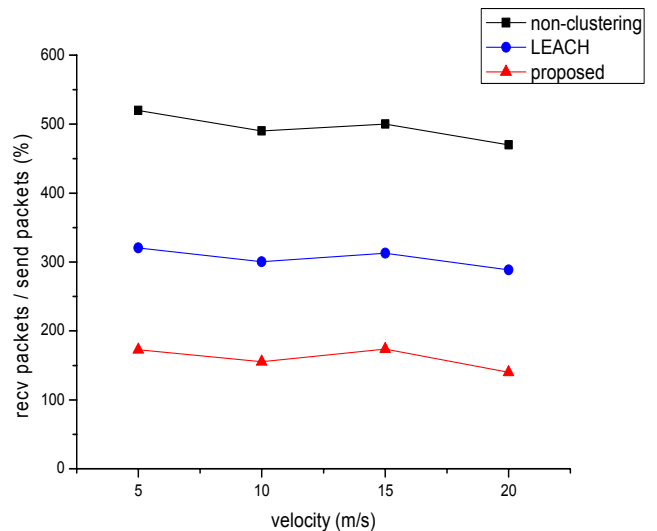


그림 10. 타깃의 이동 속도에 따른 송수신 패킷의 비율
Fig. 10. The ratios of sending and receiving packets vs. velocities for a target

V. 결 론

기존의 클러스터링 알고리즘은 이벤트 구동형 전송 모델을 고려하지 않았기 때문에 감시·정찰 응용에 적용할 경우 많은 문제점이 발생한다. 본 논문은 감시·정찰 응용에 활용될 수 있는 클러스터링 알고리즘으로 타깃의 움직임에 따라 클러스터가 생성 및 소멸 과정을 반복하고, 불필요한 control overhead를 줄여 에너지 효율을 높일 수 있는 방법을 제안하였다. 즉, 클러스터 헤드 선출 과정에서 CHEW를 이용한 지역적 경쟁 방식을 채택하고 있기 때문에, 기존 알고리즘과 같이 임의의 노드가 주변 노드들의 상태 정보를 유지하기 위한 control overhead가 발생하지 않는다. 그리고 침입자의 움직임에 민감하게 반응하여 타깃을 중심으로 클러스터가 동작하기 때문에 Boundary Problem이 발생하지 않는다. 또한, 시뮬레이션 결과를 이용하여 타깃의 이동에 따라 클러스터가 형성되는 자취를 분석하고, 에너지 효율이 증가 되는 것을 증명하였다.

본 논문에서는 감시·정찰 응용의 클러스터링 알고리즘으로 하나의 타깃을 탐지하는 것을 고려하였다. 하지만 센서 노드가 설치된 지역으로 타깃이 여러 방향에서 동시에 접근할 수도 있다. 이와 같은 상황을 고려하기 위해 본 알고리즘을 개선하여 각각의 타깃을 개별적으로 탐지하고, 그 정보를 사용자에게 손실 없이 보고하기 위한 연구를 진행할 것이다.

References

[1] Hnin Yu Shwe, JIANG xiao-hong, and susumu Horiguchi, "Energy saving in wireless sensor networks", *Journal of Communication and Computer*, vol. 6, no. 5, May 2009

[2] H. Taheri, P. Neamatollahi, M.H. Yaghmaee, and M. Naghibzadeh, "A Local Cluster Head Election Algorithm in Wireless Sensor Networks", *CSI International Symposium on Computer Science and Software Engineering(CSSE)*, pp. 38-43 June 2011

[3] S. Tilak, N. Abu-Ghazaleh and W. Heinzelman, "A Taxonomy of Wireless

Micro-Sensor Network Models", *ACM Mobile Computing and Communication Review (MC2R)*, vol. 6, no. 2, pp. 28-36, Jun. 2002

- [4] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions Wireless Communications*, vol. 1, no.4, Oct. 2002
- [5] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, "A Novel Mobility Management Scheme for Target Tracking in Cluster-Based Sensor Networks", *DISTRIBUTED COMPUTING IN SENSOR SYSTEMS Lecture Notes in Computer Science(LNCS)*, vol. 6131, pp. 172-186, Jun. 2010
- [6] O. Younis, and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", in *Proceedings of the IEEE INFOCOM*, vol. 3, pp. 629-640, 2004
- [7] Zheng Zeng-wei, Wu Zhao-hui, and Lm Huai-zhong, "An Event-Driven Clustering Routing Algorithm for Wireless Sensor Networks", in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1802-1806, Sep. 2004
- [8] O. Buyanjargal, Y. Kwon, "Adaptive and Energy Efficient clustering Algorithm for Event-Driven Application in Wireless Sensor Networks(AEEC)", *Journal of Networks*, vol. 5, no. 8, pp. 904-911, Aug. 2010
- [9] Dr.S.A Hussain, K. Zia, M.T Khan, S. Ahmad, and U. Farooq, "Dynamic Contention Window for Quality of Service in IEEE 802.11 Networks", *National Conference on Emerging Technologies 2004* pp. 12-16, Dec. 2004
- [10] <http://isi.edu/nsnam/ns>

공 준 익 (Joon-Ik Kong)



2010년 고려대 전자공학과 석사
2010년 3월~현재 고려대 전자
공학과 박사과정
<관심분야> WSN, WBAN,
Ad-Hoc, Clustering

이 재 호 (Jae-Ho Lee)



2005년 고려대 전자공학과 석사
2008년 3월~현재 고려대 전자
공학과 박사과정
2011년 3월~현재 서일대 겸임
교수
<관심분야> WPAN, 센서네트
워크, MANET, MAC, WBAN

강 지 현 (Jiheon Kang)



2008년~현재 (주) 센서웨이
2010년 3월~현재 고려대 전자
공학과 석사과정
<관심분야> 센서네트워크, MAC,
Routing, RTOS

엄 두 섭 (Doo-Seop Eom)

중신회원



1987년 고려대 전자공학과 학사
1989년 고려대 전자공학과 석사
1999년 일본 오사카대학 보통
신공학과 박사
1989년 2월~1999년 8월 한국
전자통신연구소 연구원
1999년 9월~2000년 8월 원광

대학교 전임강사

2000년 9월~현재 고려대 전기전자전파공학부 교수
<관심분야> 통신네트워크 설계 및 성능분석, 무선
ATM, IP 네트워크